Computing an Inner and an Outer Approximation of the Viability Kernel

Dominique Monnet, Jordan Ninin, Luc Jaulin

Computing an Inner and an Outer Approximation of the Viability Kernel*

D. Monnet
ENSTA-Bretagne, 2, rue François Verny, 29806 Brest Cedex 9, France
dominique.monnet@ensta-bretagne.org

J. Ninin
ENSTA-Bretagne, 2, rue François Verny, 29806 Brest Cedex 9, France
jordan.ninin@ensta-bretagne.fr

L. Jaulin
ENSTA-Bretagne, 2, rue François Verny, 29806 Brest Cedex 9, France
luc.jaulin@ensta-bretagne.fr

Abstract

The viability kernel corresponds to the set of all state vectors of a controlled dynamic system that are viable, *i.e.*, such that there exists an input such that the system will not enter inside a forbidden zone. In this paper, we propose a method which computes an inner and an outer approximation of the viability kernel in a guaranteed way. Our method is based on interval analysis and uses the notions of *V-viability* and *capture basin*. We illustrate our approach on the *car on the hill* problem. A software package has been developed to solve any 2D-problem.

Keywords: interval analysis, viability kernel
AMS subject classifications: 65-00

1 Introduction

Safety verification of controlled systems has been approached with different tools such as viability theory [2], reachability analysis, or the concept of barrier function (see *e.g.*, [5]). Safety is often expressed as a set of constraints in which the system must stay. For example, safety verification problems may consist of ensuring a safe configuration during the landing [3] or the take off [21] of an airplane, or collision avoidance [10] with other aircraft. When we have an input to the system, we may want to find a controller which guarantees that the system is safe for all conditions. The set of all state vectors

*

such that a system can stay (for the right input) within a set of constraints is called the *viability kernel*.

The computation of the viability kernel has been addressed by several methods. For low-dimensional problems and non-linear systems, most methods are based on gridding the state space [8, 20], but due to the finite precision of the computer, gridding methods cannot be considered as guaranteed. For high-dimensional systems, the viability kernel can be approximated using methods based on Lagrangian methods [15] for linear systems, or invariant sets [22] for polynomial systems. The viability kernel can also be approached using tools from the reachability analysis. Reachability analysis consists of computing the set of all state vectors that the dynamic system can reach from a given initial state. This problem has been considered with Hamilton-Jacobi equations [16], and was most recently approached with interval analysis [19]. If the target set belongs to a safe region, the viability kernel can be approximated by computing the set of state vectors from which the system can reach the target [13]. Now, for our problem of computing the viability kernel, we cannot assume a priori that such a target set, included in the viability kernel, is available. Therefore, a subset of the viability kernel must be first computed in a reliable way.

In this paper, we consider low-dimensional and non-linear dynamic systems, and we propose to compute a guaranteed approximation of the viability kernel with interval analysis tools [12]. Instead of manipulating a set of points as gridding methods do, interval methods consider connected sets of state spaces. Using interval techniques, computations are reliable and guaranteed, which allows us to compute an inner and an outer approximation of the viability kernel.

This paper is organized as follows. Section 2 contains the notation and gives the definition of the viability kernel and the capture basin. In Section 3, we present the concept of V-viability, which will allow us to find subsets of the viability kernel. Section 4 provides theoretical results to compute the capture basin and Section 5 gives an algorithm to get an inner and an outer approximation of the viability kernel. In Section 6, we illustrate our approach with a two-dimensional example. Section 7 concludes this paper.

2 Notation and Definitions

In this paper, an interval is a continuous set, denoted as follows: $[x] = [\underline{x}, \overline{x}], \underline{x} \leq \overline{x}$, with $\underline{x}$ the lower bound and $\overline{x}$ the upper bound. A vector of intervals $[x] = ([x_1], \dots, [x_q])^\intercal$ is commonly called a *box*. A dynamic system $\mathcal{S}$ is defined by the following differential equation :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), \\ u(t) \in \mathbb{U}, \end{cases} \tag{1}$$

where:

- $x(t) \in \mathbb{R}^n$ is an evolution of the state variables,
- $\mathbb{U}$ is a compact subset of $\mathbb{R}^m$ which represents the set of admissible values for the control,
- $u \in \mathcal{U} = \{u : \mathbb{R}^+ \mapsto \mathbb{U}\}$ is the control vector,
- $f : \mathbb{R}^n \times \mathbb{U} \mapsto \mathbb{R}^m$ is the evolution function of $\mathcal{S}$. We assume that f is continuous, locally Lipschitzian and bounded in $\mathbb{R}^n \times \mathbb{U}$.

Let φ be the flow map of $\mathcal{S}$, *i.e.*, with the initial condition $x_0 = x(0)$ and a control function $u(t)$, the system $\mathcal{S}$ reaches the state $\varphi(t, x_0, u)$ at time t.

Let us define the viability kernel and the capture basin of $\mathcal{S}$.

Definition 2.1 *Let $\mathcal{S}$ be a dynamic system and let $\mathbb{K} \subseteq \mathbb{R}^n$ be a compact set. The* viability kernel *of $\mathbb{K}$ under $\mathcal{S}$ is the set $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ of initial states $x \in \mathbb{K}$ from which at least one evolution does not leave $\mathbb{K}$ for all $t \geq 0$. We have*

$$\mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) = \{x \in \mathbb{K} \mid \exists u \in \mathcal{U}, \forall t \geq 0, \varphi(t, x, u) \in \mathbb{K}\}.$$

We now define the capture set (also called *robust invariant set*), which is closely related to the concept of viability kernel.

Definition 2.2 *Let $\mathbb{T} \subset \mathbb{K}$ be a target. The* capture basin *of $\mathbb{T}$ viable in $\mathbb{K}$ under $\mathcal{S}$ is the set $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T})$ of initial states $x \in \mathbb{K}$ from which at least one evolution of $\mathcal{S}$ in $\mathbb{K}$ reaches the target $\mathbb{T}$ in a finite time*

$$\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T}) = \left\{x_0 \in \mathbb{K} \mid \exists t \geq 0, \exists u \in \mathcal{U}, \ \begin{cases} \varphi(t, x_0, u) \in \mathbb{T}, \\ \varphi([0, t], x_0, u) \subseteq \mathbb{K}. \end{cases} \right\}.$$

where

$$\varphi([t_1, t_2], x_0, u) = \{x \in \mathbb{R}^n \mid \exists t \in [t_1, t_2], x = \varphi(t, x_0, u)\}. \tag{2}$$

3 V-Viability

Definition 3.1 *Consider a dynamic system $\mathcal{S}$ and a differentiable function $V : \mathbb{R}^n \mapsto \mathbb{R}$. $\mathcal{S}$ is said to be V-viable if*

$$\forall x \in \mathbb{R}^n \text{ such that } V(x) = 0, \exists u \in \mathbb{U}, \langle \nabla V(x) \cdot f(x, u) \rangle < 0.$$

Figure 1 illustrates a V-viable system. For each point of the curve $V(x) = 0$ at least one potential evolution of $\mathcal{S}$ points strictly inward into the gray set, which represents $V^{-1}(\mathbb{R}^-)$. This is the geometrical interpretation of $\langle \nabla V(x) \cdot f(x, u) \rangle < 0$.

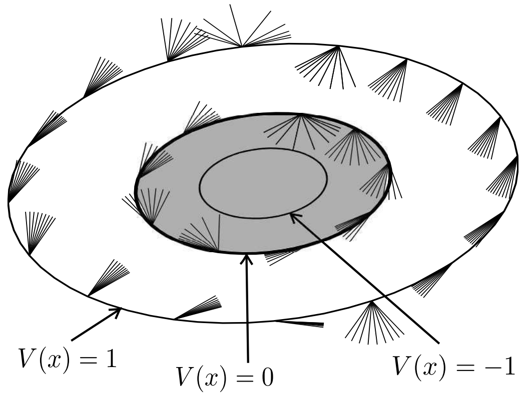


Figure 1: The vector field is displayed for several controls along the level curves of V.

The definition of V-viability implies the resolution of a quantified constraint satisfaction problem that can be solved with cylindrical decomposition [6] if V is polynomial or with methods based on interval analysis [11] in the general case.

Theorem 3.1 *Consider $\mathbb{K}$ a subset of $\mathbb{R}^n$. If $\mathcal{S}$ is V-viable and $V^{-1}(\mathbb{R}^-)$ is a compact subset of $\mathbb{K}$, then $V^{-1}(\mathbb{R}^-) \subseteq \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.*

Proof: Consider $x \in \mathbb{K}$ and the subset $\mathbb{L}$ of $\mathbb{K}$ defined by

$$\mathbb{L} = \{x \in \mathbb{K} | V(x) \leq 0\} = V^{-1}(\mathbb{R}^-)$$

The contingent cone $T_{\mathbb{L}}(x)$ to $\mathbb{L}$ at $x \in \mathbb{L}, V(x) = 0$ can be defined with the gradient of V. According to [1, pp. 123], $T_{\mathbb{L}}(x)$ is defined by

$$T_{\mathbb{L}}(x) = \{y \in \mathbb{K} |\, V(x) = 0 \implies \langle \nabla V(x), y \rangle \leq 0\}$$

Here we recall the definition of a viability domain [2, pp. 84]:
We shall say that a subset $\mathbb{L} \subset \mathbb{K}$ is a viability domain *of $\mathbb{K}$ if and only if*

$$\forall x \in \mathbb{L},\ f(x, \mathbb{U}) \cap T_{\mathbb{L}}(x) \neq \emptyset$$

As $\mathcal{S}$ is V-viable, we have:

$$\begin{aligned} &\forall x \text{ such as } V(x) = 0,\ \exists u \in \mathbb{U},\ \langle V(x), f(x, u) \rangle < 0 \\ &\qquad\qquad\qquad\qquad \implies f(x, \mathbb{U}) \cap T_{\mathbb{L}}(x) \neq \emptyset \end{aligned}$$

Therefore, $\mathbb{L}$ is a viability domain.
Moreover, according to the Local viability theorem [2, pp. 91], we have:

$$x_0 \in \mathbb{L} \implies \exists u \in \mathcal{U} \text{ such as } \forall t \geq 0,\ \varphi(x_0, t, u) \in \mathbb{K}.$$

Thus, by definition, $\mathbb{L} \subseteq \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. □

Theorem 3.1 can be used to prove that a set is inside the viability kernel of $\mathbb{K}$. Figure 1 illustrates this by the gray set, which belongs to $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. To find a viable subset of the viability kernel, a function V must be chosen so that $\mathcal{S}$ has a high probability to be V-viable. Such a function can be found using Lyapunov theory [4, 23]. Some Lyapunov methods can approximate attraction domains of $\mathcal{S}$ [9, 18]. If $\mathcal{S}$ is V-viable and the second condition of Theorem 3.1 is checked, $V^{-1}(\mathbb{R}^-)$ is proved to be a subset of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.

4 Capture Basin of Viable Sets

In Section 3, a method to compute subsets of the viability kernel from the notion of V-viability is provided. These subsets represent a first approximation of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. In order to obtain a better approximation, these viable subsets are enlarged with their capture basins.

The following theorem states that the capture basin of a subset of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ is also a subset of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.

Theorem 4.1 *Let $\mathcal{S}$ be a dynamic system, $\mathbb{K}$ be a closed subset of the state space of $\mathcal{S}$, and $\mathbb{T}$ be a subset of $\mathbb{K}$. If $\mathbb{T} \subset \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$, then $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T}) \subset \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.*

Proof: Consider $x_0 \in \mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T})$. From the definition of $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T})$, we have

$$\exists t_1 \geq 0, \exists u_1 \in \mathcal{U}, \varphi(t_1, x_0, u_1) \in \mathbb{T} \text{ and } \varphi([0, t_1], x_0, u_1) \subset \mathbb{K}.$$

Moreover, if $\mathbb{T} \subseteq \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ then $\exists u_2 \in \mathcal{U}, \forall t > t_1, \varphi(t, \varphi(t_1, x_0, u_1), u_2) \in \mathbb{K}$. Define

$$u(t) = \begin{cases} u_1(t) & \text{if } t \in [0, t_1], \\ u_2(t) & \text{if } t > t_1. \end{cases} \tag{3}$$

Then, $\forall t > 0$, $\varphi(t, x_0, u) \in \mathbb{K}$. So, according to the definition of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$, we have $x_0 \in \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. □

Proposition 4.1 *If $\mathbb{T} \subset \mathbb{K}$, then*

(i) $\exists t \geq 0, \exists u \in \mathcal{U}, \begin{cases} \varphi(t, x_0, u) \in \mathbb{T} \\ \varphi([0, t], x_0, u) \subseteq \mathbb{K} \end{cases} \implies x_0 \in \mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T})$,

(ii) $\exists t \geq 0, \forall u \in \mathcal{U}$, $\varphi(t, x_0, u) \notin \mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) \implies x_0 \notin \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$,

(iii) $x_0 \notin \mathbb{K} \implies x_0 \notin \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.

Proof: This proof is trivial since it comes from the definition of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. □

Proposition 4.1 may be used to prove that a state belongs to $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{E})$ or does not belong to $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.

Remark 4.1 *From Equation (iii) of Proposition 4.1, we can conclude that* $(\mathbb{R}^n \setminus \mathbb{K}) \cap \mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) = \emptyset$.

In the following, we consider an initial box $[x_0]$ instead of a unique point state.

Proposition 4.2 *Let $[x_0] \subset \mathbb{K}$ be an initial set and $\mathbb{T} \subset \mathbb{K}$*

(i) if $\forall x \in [x_0]$, $\exists t \geq 0, \exists u \in \mathcal{U}$, $\varphi(t, x, u) \subset \mathbb{T}$ *and* $\varphi([0, t], x, u) \subseteq \mathbb{K}$,
then $[x_0] \subset \mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{T})$,

(ii) if $\forall x \in [x_0]$, $\exists t \geq 0, \forall u \in \mathcal{U}$, $\varphi(t, x, u) \notin \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$,
then $[x_0] \cap \mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) = \emptyset$.

Proof: The proof of *(i)* and *(ii)* are the same of those of Proposition 4.1 considering sets of states variable instead of unique point. □

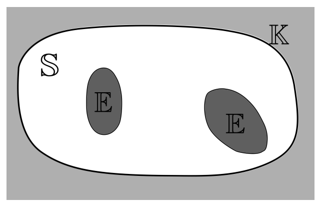
Methods using interval arithmetic are able to compute a guaranteed enclosure of the flow map φ from an initial box $[x_0]$ of the state space, see [7, 17] for details. Using these techniques, it is possible to compute

- an over-approximation $[\varphi](t, [x_0], u)$ of $\{\varphi(t, x, u) \mid x \in [x_0]\}$,
- an enclosure $[\varphi]([0, t], [x_0], u)$ of all evolutions $\varphi([0, t], x, u)$ with all initial states $x \in [x_0]$,
- an enclosure $[\varphi](t, [x_0], \mathcal{U})$ of all evolutions $\varphi(t, x, u)$ with all initial states $x \in [x_0]$ and all the possible control functions $u \in \mathcal{U}$.

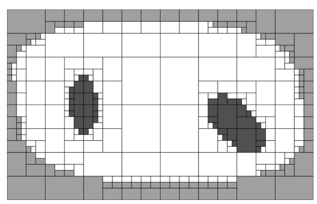
Such enclosures are generally overestimated. But if the size of the box $[x_0]$ is small enough, this estimation can be accurate.

5 Algorithm to Compute $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$

In this section, we suppose that we have found a set $\mathbb{E} \subseteq \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ using techniques of Section 3. We present an algorithm that computes $\mathbb{V}_{\text{inner}}$, a guaranteed inner approximation of $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{E})$, and $\mathbb{H}$, a guaranteed approximation of the complement of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ in $\mathbb{K}$. This algorithm is an extension of the one presented in [14] to compute the viability kernel. We choose to subdivide the state space into boxes, so that we can easily use Proposition 4.2. Figure 2a shows the initial problem. Its representation with a subpaving is displayed in Figure 2b.



(a)



(b)

Figure 2: Subpaving representing the initial inner approximation for $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$

$\mathbb{V}_{\text{inner}}$ is initialized with boxes of the subpaving proved to belong to $\mathbb{E}$, represented in dark gray in Figure 2b. $\mathbb{H}$ is initially empty. We want to check whether boxes contained in the set $\mathbb{S} = \mathbb{K} \setminus (\mathbb{H} \cup \mathbb{V}_{\text{inner}})$ in white on Figure 2b belong to $\mathbb{V}_{\text{inner}}$ or $\mathbb{H}$. The gray space represents the complement of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ in $\mathbb{R}^n$ and is proved not to belong to $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ (see Remark 4.1). Figure 3 illustrates the principle of Algorithm 1.

Algorithm 1 works as follows. A control u is randomly picked in $\mathcal{U}$. The value t is fixed at the beginning of Algorithm 1. If Equation *(i)* of Proposition 4.2 holds (Line 4), $[x_i]$ is proved to belong to $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{E})$ and we add it to $\mathbb{V}_{\text{inner}}$. This case is represented by $[x_1]$ in Figure 3. The box $[x_2]$ represents a case in which nothing can be proved. We know that $((\mathbb{R}^n \setminus \mathbb{K}) \cup \mathbb{H}) \cap \mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) = \emptyset$. If the condition of Line 6 is checked, we have $[\varphi](t, [x_i], \mathcal{U}) \cap \mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) = \emptyset$. Thus, from Equation *(ii)* of Proposition 4.2, we know that $[x_i]$ does not intersect $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$ and we add it to $\mathbb{H}$. The box $[x_3]$ illustrates this case on Figure 3. Nothing can be proved for $[x_4]$. At Line 10, $\mathbb{S}$ contains boxes that we have not proved whether they belong to $\mathbb{H}$ or $\mathbb{V}_{\text{inner}}$. These boxes are bisected to facilitate the classification with respect to $\mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{E})$ or $\mathbb{H}$.

At the end of Algorithm 1, we have $\mathbb{V}_{\text{inner}} \subseteq \mathrm{Capt}_{\mathcal{S}}(\mathbb{K}, \mathbb{E})$. Moreover, according to Theorem 4.1, we have $\mathrm{Capt}_{\mathcal{S}}(\mathbb{E}, \mathbb{K}) \subseteq \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. Hence $\mathbb{V}_{\text{inner}} \subseteq \mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$. Finally, $\mathbb{V}_{\text{inner}}$ is an inner approximation of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$.

Furthermore, $\mathbb{H} \cap \mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) = \emptyset$, thus we have $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K}) \subseteq (\mathbb{K} \setminus \mathbb{H})$. Then, the

Algorithm 1 Computation of an inner and an outer approximation of $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$

Require: Initial sets $\mathbb{S}$,$\mathbb{V}_{\text{inner}}$ and $\mathbb{H}$.

while $\mathbb{S} \neq \emptyset$ **do**
    for all $[x_i]$ in $\mathbb{S}$ **do**
        Choose $u \in \mathcal{U}$.
        if $[\varphi]([0,t],[x_i],u) \subseteq \mathbb{K}$ and $[\varphi](t,[x_i],u) \subseteq \mathbb{V}_{\text{inner}}$ **then**
            $\mathbb{V}_{\text{inner}} := \mathbb{V}_{\text{inner}} \cup [x_i], \quad \mathbb{S} := \mathbb{S} \setminus [x_i],$
        else if $[\varphi](t,[x_i],\mathbb{U}) \subseteq ((\mathbb{R}^n \setminus \mathbb{K}) \cup \mathbb{H})$ **then**
            $\mathbb{H} := \mathbb{H} \cup [x_i], \quad \mathbb{S} := \mathbb{S} \setminus [x_i],$
        end if
    end for
    Bisect boxes of $\mathbb{S}$.
end while
return $\mathbb{V}_{\text{inner}}$ and $\mathbb{H}$.

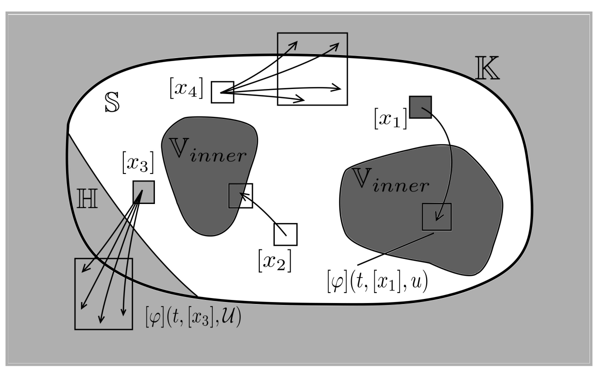


Figure 3: Illustration of the method for characterizing $\mathrm{Viab}_{\mathcal{S}}(\mathbb{K})$

set $\mathbb{V}_{\text{outer}} = \mathbb{K} \setminus \mathbb{H}$ is an outer approximation of $\text{Viab}_{\mathcal{S}}(\mathbb{K})$. Due to the reliable and guaranteed computation of $[\varphi]$ with interval arithmetic, the inner and outer approximations $\mathbb{V}_{\text{inner}}$ and $\mathbb{V}_{\text{outer}}$ are certified.

Compared to the algorithm of [14], our version contains technical and computational improvements to accelerate the convergence:

- The enclosure of $[\varphi]$ is computed using the technique of [7], which improves the quality of the bounds and increases the chance to include a box in $\mathbb{V}_{\text{inner}}$ or $\mathbb{H}$.
- The data structure of $\mathbb{V}_{\text{inner}}$ and $\mathbb{H}$ is carefully implemented. A regular paver, represented by a binary tree, is used to perform fast intersections and set unions .
- In Algorithm 1, the subboxes composing $\mathbb{S}$ are divided only if it is not possible to prove that they belong to $\mathbb{V}_{\text{inner}}$ or $\mathbb{H}$. In the algorithm presented in [14], $\mathbb{S}$ is entirely dividied into small boxes with the same size, then it tries to prove if they belong to $\mathbb{V}_{\text{inner}}$ or $\mathbb{H}$. Algorithm 1 divides boxes only if needed; this allows it to include large boxes directly in $\mathbb{V}_{\text{inner}}$ or $\mathbb{H}$.

These points aim to break down the exponential complexity of this algorithm.

6 Application to the *Car on the Hill* Problem

We illustrate our approach with the *car on the hill* problem. In this application, we want the car to stay on a landscape represented by the parametric function

$$g : s \mapsto \frac{\frac{-1.1}{1.2}\cos(1.2s) + \frac{1.2}{1.1}\cos(1.1s)}{2},$$

where $s \in [0, 12]$ denotes the longitudinal variable. The function g is plotted on Figure 4.

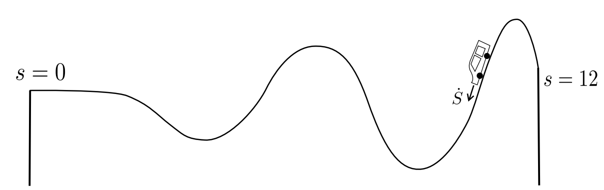


Figure 4: Car on the hill problem. We aimed to avoid the car falling off the cliffs at $s = 0$ and $s = 12$.

The acceleration of the car can be controlled within a limited range. We also consider a friction force that slows down the car. The dynamic system $\mathcal{S}$ associated with our problem is described by

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = -9.81\sin(\dot{g}(x_1(t))) - 0.7x_2(t) + u(t), \end{cases}$$

where $x_1 = s$ represents the position of the car on the landscape, $x_2 = \dot{s}$ its velocity, and $u \in [-2, 2]$ is the control function. The viability problem is formulated as follows:

we want to keep x_1 between values 0 and 12 with a limit on the velocity of the car between -6 *and* 6. Thus, the set of constraints $\mathbb{K}$ is defined by

$$\mathbb{K} = \{x \mid x_1 \in [0, 12], x_2 \in [-6, 6]\}.$$

We limit the precision on s and $\dot{s}$ to 0.1.
To compute $\text{Viab}_{\mathcal{S}}(\mathbb{K})$ we first apply the method of Section 3 to find a subset $\mathbb{E}$ of $\text{Viab}_{\mathcal{S}}(\mathbb{K})$. To do so, $\mathcal{S}$ is linearized around an equilibrium point of $\mathcal{S}$. Then, a quadratic Lapunov function is computed for the linearized system. A function V is created from the Lyapunov function so that $\mathcal{S}$ is V-viable. This procedure is repeated for several equilibrium points. Figure 5 illustrates the result obtained.
The elliptic shapes of subsets of $\text{Viab}_{\mathcal{S}}(\mathbb{K})$ in Figure 5 come from the quadratic expressions of the Lyapunov functions. Moreover, the centers of these viable subsets correspond to a position of the car at the tops or bottoms of the hills with a null velocity, where the car is in an equilibrium state.

The computation took 10 seconds on a 2.5 GHz Intel Core i5-2450M processor with 6 Gb RAM.

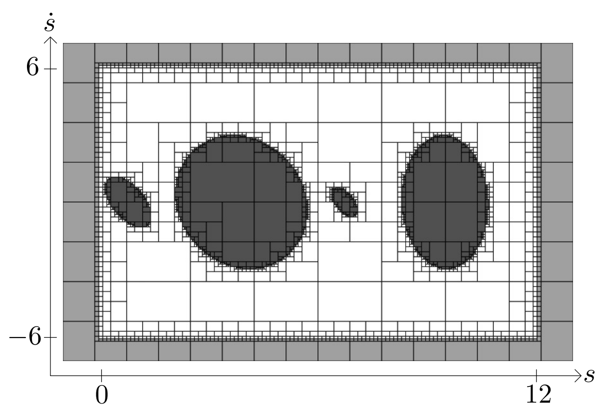


Figure 5: Subsets of $\text{Viab}_{\mathcal{S}}(\mathbb{K})$ are found with method of Section 3

Next, we apply Algorithm 1 to compute an inner and outer approximation of $\text{Viab}_{\mathcal{S}}(\mathbb{K})$. The result is shown in Figure 6. The CPU time is 1h 23min. The dark-gray set corresponds to $\mathbb{V}_{\text{inner}}$ and is the largest set that we could prove to be viable in a guaranteed way. Nothing could be proved for the boxes of the white set. The light gray set is proved to be outside the viability kernel. The union of the white set and the dark-gray set corresponds to $\mathbb{V}_{\text{outer}}$.
Figures 5 and 6 have been obtained with a solver available at
`http://www.ensta-bretagne.fr/monnet/Viabibex/`.

7 Conclusion

This paper proposes an interval method to approximate the viability kernel of a non-linear dynamic system. Our approach merges the concepts of *V-viability* and reachability analysis. An inner approximation of the viability kernel is computed on an

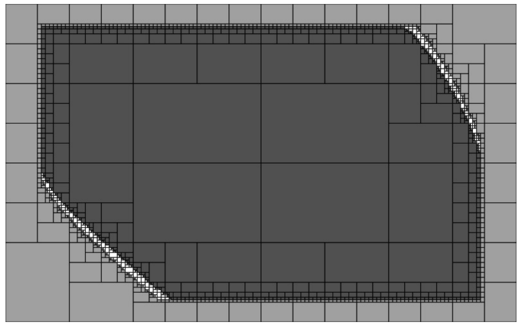


Figure 6: Approximations of the viability kernel.

infinite time horizon. Interval analysis tools ensure the reliability and provide a guarantee for all the results. The non-linear constraints on the evolution function enable application of our method to a large variety of problems. Numerical results for the *car on the hill* problem demonstrate the feasibility of our approach. Moreover, our method can be generalized to an n-dimensional problem.

References

[1] J. P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhuser Boston, 1990.

[2] Jean-Pierre Aubin. *Viability Theory*. Birkhäuser Boston, 1991.

[3] A. M. Bayen, I. M. Mitchell, M. K. Osihi, and C. J. Tomlin. Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation. *Journal of Guidance, Control, and Dynamics*, 30(1):68–77, 2007.

[4] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Springer Science & Business Media, October 2007.

[5] O. Bouissou, A. Chapoutot, A. Djaballah, and M. Kieffer. Computation of parametric barrier functions for dynamical systems using interval analysis. In *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 753–758, December 2014.

[6] B. F. Caviness and J. R. Johnson. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer Science & Business Media, December 2012.

[7] A. Chapoutot, J. Alexandre Dit Sandretto, and O. Mullier. Validated explicit and implicit Runge-Kutta methods. *Reliable Computing*, 22:79–103, July 2016. Proceedings of the Small Workshop on Interval Methods, June, 2015.

[8] G. Deffuant, L. Chapel, and S. Martin. Approximating Viability Kernels With Support Vector Machines. *IEEE Transactions on Automatic Control*, 52(5):933–937, May 2007.

[9] N. Delanoue, L. Jaulin, and B. Cottenceau. An algorithm for computing a neighborhood included in the attraction domain of an asymptotically stable point. *Communications in Nonlinear Science and Numerical Simulation*, 21(13):181–189, April 2015.

[10] A. Desilles, H. Zidani, and E. Crck. Collision analysis for an UAV. In *AIAA Guidance, Navigation, and Control Conference*, pages 13–16. American Institute of Aeronautics and Astronautics, 2012.

[11] M. Hladík and S. Ratschan. Efficient solution of a class of quantified constraints with quantifier prefix Exists-Forall. *Mathematics in Computer Science*, 8(3-4):329–340, July 2014.

[12] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, UK, 2001.

[13] S. Kaynama, J. Maidens, M. Oishi, I. M. Mitchell, and G. A. Dumont. Computing the Viability Kernel Using Maximal Reachable Sets. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '12, pages 55–64, New York, NY, USA, 2012. ACM.

[14] M. Lhommeau, L. Jaulin, and L. Hardouin. Capture basin approximation using interval analysis. *International Journal of Adaptive Control and Signal Processing*, 25(3):264–272, March 2011.

[15] J. N. Maidens, S. Kaynama, . M. Mitchell, M. Oishi, and G. A. Dumont. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica*, 49(7):2017–2029, July 2013.

[16] Ian Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. Validating a Hamilton-Jacobi Approximation to Hybrid System Reachable Sets. In M. Domenica Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in Lecture Notes in Computer Science, pages 418–432. Springer Berlin Heidelberg, 2001.

[17] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105(1):21–68, October 1999.

[18] S. Ratschan and Z. She. Providing a basin of attraction to a target region of polynomial systems by computation of Lyapunov-like functions. *SIAM Journal on Control and Optimization*, 48(7):4377–4394, January 2010.

[19] F. Rego, E. de Weerdt, E. van Oort, E. van Kampen, Q. Chu, and A. M. Pascoal. Determination of inner and outer bounds of reachable sets through subpavings. *Mathematics in Computer Science*, 8(3-4):425–442, August 2014.

[20] P. Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29(2):187–209, March 1994.

[21] N. Seube, R. Moitie, and G. Leitmann. Aircraft take-off in windshear: A viability approach. *Set-Valued Analysis*, 8(1-2):163–180, June 2000.

[22] Z. She and B. Xue. Brief paper: Computing an invariance kernel with target by computing Lyapunov-like functions. *IET Control Theory Applications*, 7(15):1932–1940, October 2013.

[23] J. J. Slotine and W. Li. *Applied Nonlinear Control*, volume 199. Prentice-hall Englewood Cliffs, NJ, 1991.